REMARKS

The outstanding non-final Office Action mailed January 6, 2005 has been carefully considered. In response thereto, Applicants submit herewith the foregoing amendments and following remarks. Claims 1 – 24 remain pending. Claims 1, 9, and 15 are amended. The subject matter of amended claims 1, 9, and 15 can be found in Applicants' originally submitted drawings and the related detailed description. For example, see at least page 25, lines 13 – 22 and page 26, lines 22 – 25 of Applicants' originally filed specification. Thus, Applicants submit no new matter has been added to the application.

In view of the foregoing amendments and the following remarks, reconsideration and allowance of the present application and claims 1 - 24 are respectfully requested.

I. Claim Rejections under 35 U.S.C. § 102(e) - Claims 1 – 4 and 6 – 24

A. Statement of the Rejection

The Office Action indicates that claims 1 – 4 and 6 – 24 stand rejected under 35 U.S.C. §102(e) as allegedly being anticipated by U.S. Patent Publication No. 2002/0116635 to Sheymov *et al.*, hereafter *Sheymov*.

B. Discussion of the Rejection - Claims 1 – 4 and 6 – 24

Applicants' claims are not anticipated for at least the reason that the cited reference fails to disclose, teach, or suggest each element in the claims.

It is axiomatic that "[a]nticipation requires the disclosure in a single prior art reference of *each element* of the claim under consideration." *W.L. Gore & Associates, Inc. v. Garlock, Inc.*, 721 F.2d 1540, 1554, 220 U.S.P.Q. 303, 313 (Fed. Cir. 1983) (*emphasis added*). Therefore, every claimed feature of the claimed invention must be represented in the applied reference (*i.e.*, *Greidinger*) to constitute a proper rejection under 35 U.S.C. §102(e).

1. Claims 1 – 4 and 6 – 8

Applicants' independent claim 1 is exemplary. For convenience of analysis, independent claim 1, as amended, is repeated on the following page in its entirety.

> 1. A method for identifying infected program instructions, comprising the steps of:
>
> inserting a dynamic execution layer interface (DELI) between computing device hardware and the program instructions;
>
> ***monitoring the program instructions as they enter the DELI to determine if the program instructions have been previously cached within the DELI, wherein the determination of whether the program instructions have been cached is responsive to an association between native application code and one or more analogues that have been transformed within the DELI***; and when it is the case that the program instructions have not been previously cached within the DELI,
>
> analyzing the program instructions to determine if the program instructions are infected.

(Applicants' independent claim 1 – *Emphasis added.*)

Applicants respectfully assert that the cited art of record fails to disclose, teach, or suggest at least the emphasized feature of pending claim 1 as shown above. Consequently, claim 1 is allowable.

Specifically, *Sheymov* fails to disclose, teach, or suggest a method for identifying infected program instructions that comprises "monitoring the program instructions as they enter the DELI to determine if the program instructions have been previously cached within the DELI, wherein the determination of whether the program instructions have been cached is responsive to an association between native application code and one or more analogues that have been transformed within the DELI." *Sheymov* is entirely silent regarding determining whether program instructions have been previously cached within a dynamic execution layer interface.

In contrast with Applicants' claimed method, *Sheymov* apparently discloses a system that determines when a protected system is new. In accordance with the cited portion of *Sheymov*, a protected system is new and thus analyzed, when the system has a new component and/or a new application has been installed. In response, *Sheymov* describes initializing actuator and sensor modules that reside in a dynamic decoy machine that mimics operation of the protected system. The actuator emulates normal use of the protected machine, *i.e.*, it opens and closes applications, accesses files, sends various communications, manipulates user-definable parameter values, or the like, and accelerates the clock of the decoy machine to trigger time-triggered malicious codes. Access based sensors detect

attempts to access undesirable areas of the decoy machine addition. Accordingly, *Sheymov* describes simply copying environmental parameters and applications and emulating operation of a protected system.

Applicants' claimed method includes monitoring the program instructions as they enter the DELI to determine if the program instructions have been previously cached within the DELI, wherein the determination of whether the program instructions have been cached is responsive to an association between native application code and one or more analogues that have been transformed within the DELI. *Sheymov* does not disclose, teach, or suggest that program instructions should be monitored to determine if they have been previously cached within a dynamic execution layer interface between the program instructions and computer hardware. Accordingly, for at least this reason, *Sheymov* does not anticipate Applicants' claimed method. Consequently, claim 1 is allowable and the rejection of claim 1 should be withdrawn.

Furthermore, Applicants' method includes the feature that the determination of whether the program instructions have been cached is responsive to an association between native application code and analogues that have been transformed within the DELI. *Sheymov* fails to disclose, teach, or suggest any transformation of application code within the decoy machine. Accordingly, for at least this separate reason, *Sheymov* does not anticipate Applicants' claimed method.

Because independent claim 1 is allowable, dependent claims 2 – 4 and 6 – 8, which depend from claim 1, are also allowable. *See In re Fine*, 837, F.2d 1071, 5 U.S.P.Q.2d 1596, 1598. (Fed. Cir. 1988). Accordingly, Applicants respectfully request that the rejection of claims 1 – 4 and 6 – 8 be withdrawn.

2. Claims 9 – 14

Applicants' independent claim 9 is also allowable. For convenience of analysis, independent claim 9, as amended, is repeated below in its entirety.

> 9. A system for detecting infected program instructions in active software applications, comprising:
> means for intercepting program instructions designated for execution within a computing device;
> ***means for transforming the program instructions***;
> ***means for determining when the intercepted program instructions have not been processed by the computing device***

> ***responsive to an association between native application code from the active software applications and one or more analogues that have been cached within a dynamic execution layer inserted between a processor and program instructions***; and
>
> means for analyzing the intercepted program instructions that have not been processed by the computing device prior to forwarding the intercepted program instructions to computer hardware.

(Applicants' independent claim 9 – *Emphasis added.*)

Applicants respectfully assert that the cited art of record fails to disclose, teach, or suggest at least the emphasized features of pending claim 9 as shown above. Consequently, claim 9 is allowable.

Specifically, *Sheymov* fails to disclose, teach, or suggest a system for detecting infected program instructions in active software applications that comprises "means for transforming the program instructions;" and "means for determining when the intercepted program instructions have not been processed by the computing device responsive to an association between native application code from the active software applications and one or more analogues that have been cached within a dynamic execution layer inserted between a processor and program instructions." As shown above, *Sheymov* is entirely silent regarding determining whether program instructions have been previously cached within a dynamic execution layer interface. In addition, *Sheymov* is entirely silent regarding transforming the program instructions.

In contrast with Applicants' claimed system, *Sheymov* apparently describes initializing actuator and sensor modules that reside in a dynamic decoy machine that mimics operation of the protected system. The actuator emulates normal use of the protected machine, *i.e.*, it opens and closes applications, accesses files, sends various communications, manipulates user-definable parameter values, or the like, and accelerates the clock of the decoy machine to trigger time-triggered malicious codes. Access based sensors detect attempts to access undesirable areas of the decoy machine. Accordingly, *Sheymov* describes simply copying environmental parameters and applications and emulating operation of a protected system.

Applicants' claimed system for detecting infected program instructions in active software applications includes means for transforming the program instructions. *Sheymov* does not disclose, teach, or suggest that program instructions should transformed in any way.

Accordingly, for at least this reason, *Sheymov* does not anticipate Applicants' claimed system. Consequently, claim 9 is allowable and the rejection of claim 9 should be withdrawn.

Furthermore, Applicants' claimed system for detecting infected program instructions in active software applications includes a means for determining when the intercepted program instructions have not been processed by the computing device responsive to an association between native application code from the active software applications and one or more analogues that have been cached within a dynamic execution layer inserted between a processor and program instructions. *Sheymov* fails to disclose, teach, or suggest that program instructions should be analyzed to determine if they have been previously cached by a dynamic execution layer inserted between a processor and program instructions. Accordingly, for at least this separate reason, *Sheymov* does not anticipate Applicants' claimed system.

Because independent claim 9 is allowable, dependent claims 10 – 14, which depend from claim 9, are also allowable. *See In re Fine, supra.* Accordingly, Applicants respectfully request that the rejection of claims 9 – 14 be withdrawn.

3. Claims 15 - 18

Applicants' independent claim 15 is also allowable. For convenience of analysis, independent claim 15, as amended, is repeated below.

> 15. A virus detection program stored on a computer-readable medium, comprising:
>
> logic configured to intercept program instructions;
>
> ***logic configured to transform the program instructions***;
>
> ***logic configured to determine if the intercepted program instructions have not been processed by a computing device responsive to an association between application code and one or more analogues that have been cached within a dynamic execution layer inserted between a processor and program instructions***; and
>
> logic configured to determine when the intercepted program instructions that have not been processed by the computing device are infected with a virus.

(Applicants' independent claim 15 – *Emphasis added.*)

Applicants respectfully assert that the cited art of record fails to disclose, teach, or suggest at least the emphasized features of Applicants' claim 15 as shown above. Consequently, claim 15 is allowable.

Specifically, *Sheymov* fails to disclose, teach, or suggest a virus detection program stored on a computer-readable medium that comprises "logic configured to transform the program instructions;" and "logic configured to determine if the intercepted program instructions have not been processed by a computing device responsive to an association between application code and one or more analogues that have been cached within a dynamic execution layer inserted between a processor and program instructions." As shown above, *Sheymov* is entirely silent regarding transforming program instructions. In addition, *Sheymov* is entirely silent regarding determining if the intercepted program instructions have not been processed by a computing device responsive to an association between application code and analogues that have been cached within code caches within a dynamic execution layer inserted between a processor and program instructions.

In contrast with Applicants' claimed virus detection program, *Sheymov* apparently describes initializing actuator and sensor modules that reside in a dynamic decoy machine that mimics operation of the protected system. The actuator emulates normal use of the protected machine, *i.e.*, it opens and closes applications, accesses files, sends various communications, manipulates user-definable parameter values, or the like, and accelerates the clock of the decoy machine to trigger time-triggered malicious codes. Access based sensors detect attempts to access undesirable areas of the decoy machine. Accordingly, *Sheymov* describes simply copying environmental parameters and applications and emulating operation of a protected system.

Applicants' claimed virus detection program includes logic configured to transform the program instructions. *Sheymov* does not disclose, teach, or suggest that program instructions should transformed in any way. Accordingly, for at least this reason, *Sheymov* does not anticipate Applicants' claimed virus detection program. Consequently, claim 15 is allowable and the rejection of claim 15 should be withdrawn.

Furthermore, Applicants' claimed virus detection program includes logic configured to determine if the intercepted program instructions have not been processed by a computing device responsive to an association between application code and one or more analogues that have been cached within a dynamic execution layer inserted between a processor and program

instructions. *Sheymov* fails to disclose, teach, or suggest that program instructions should be analyzed to determine if they have not been processed by a computing device responsive to an association between application code and one or more analogues that have been cached within a dynamic execution layer inserted between a processor and program instructions. Accordingly, for at least this separate reason, *Sheymov* does not anticipate Applicants' claimed virus detection program.

Because independent claim 15 is allowable, dependent claims 16 – 18, which depend from claim 15, are also allowable. *See In re Fine, supra.* Accordingly, Applicants respectfully request that the rejection of claims 15 – 18 be withdrawn.

4. Claims 19 – 24

Applicants respectfully traverse the rejection of Applicants' originally filed independent claim 19. For convenience of analysis, independent claim 19, as originally filed, is repeated below in its entirety.

> 19. A computer system, comprising:
> a processor;
> an execution memory;
> ***a dynamic execution layer interface (DELI) residing between at least one application and the processor***, wherein the DELI comprises:
> a core configured to cache and execute certain application code fragments;
> ***an application programming interface configured to provide access to caching and executing functions of the core to a virus detection manager***; and
> ***a system control and configuration layer configured to provide policies for operation of the core.***

(Applicants' independent claim 19 – *Emphasis added.*)

Applicants respectfully assert that the cited art of record fails to disclose, teach, or suggest at least the emphasized features of Applicants' claim 19 as shown above. Consequently, claim 19 is allowable.

Specifically, *Sheymov* fails to disclose, teach, or suggest a dynamic execution layer interface residing between at least one application and the processor wherein the interface comprises "an application programming interface configured to provide access to caching

and executing functions of the core to a virus detection manager;" and "a system control and configuration layer configured to provide policies for operation of the core."

Sheymov apparently describes initializing actuator and sensor modules that reside in a dynamic decoy machine that mimics operation of the protected system. *Sheymov* is entirely silent regarding a dynamic execution layer interface residing between at least one application and the processor wherein the interface comprises "an application programming interface configured to provide access to caching and executing functions of the core to a virus detection manager;" and "a system control and configuration layer configured to provide policies for operation of the core." Accordingly, claim 19 is not anticipated by *Sheymov*.

Because independent claim 19 is allowable, dependent claims 20 – 24, which depend from claim 19, are also allowable. *See In re Fine, supra*. Accordingly, Applicants respectfully request that the rejection of claims 19 – 24 be withdrawn.

II. Claim Rejections under 35 U.S.C. §103 - Claim 5

A. Statement of the Rejections

The Office Action indicates that claim 5 is rejected under 35 U.S.C. §103(a) as being unpatentable over *Sheymov* in view of U.S. Patent No. 6,577,920 to Hyppönen *et al.* (hereafter *Hyppönen*).

B. Discussion of the Rejections

Applicants' dependent claim 5 includes features that are not disclosed, taught, or suggested by the proposed combinations of *Sheymov* and *Hyppönen*.

To establish a *prima facie* case of obviousness based on a combination of the content of various references, there must be some teaching, suggestion or motivation in the prior art to make the specific combination that was made by the applicant. *In re Raynes*, 7 F.3d 1037, 1039, 28 USPQ2d 1630, 1631 (Fed. Cir. 1993); *In re Oetiker*, 977 F.2d 1443, 1445, 24 USPQ2d 1443, 1445 (Fed. Cir. 1992). As stated in MPEP 2143 - Basic Requirements of a *Prima Facie* Case of Obviousness,

> [t]o establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference

> (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art and not based on applicant's disclosure. *In re Vaeck,* 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991).

Applicants' dependent claim 5 includes claim limitations that are not disclosed, taught, or suggested by the cited references. Accordingly, the proposed combinations fail to establish a *prima facie* case for obviousness for failure to teach or suggest all the claim limitations.

Specifically, the proposed combination fails to disclose, teach, or suggest "monitoring the program instructions as they enter the DELI to determine if the program instructions have been previously cached within the DELI, wherein the determination of whether the program instructions have been cached is responsive to an association between native application code and analogues that have been cached within the code caches within the DELI." In this regard, both *Sheymov* and *Hyppönen* are entirely silent.

Hyppönen apparently describes a method of screening a software file for a viral infection comprising defining a first database of known macro virus signatures, a second database of known and certified commercial macro signatures, and a third database of known and certified local macro signatures. When a file contains a macro, *Hyppönen* screens the macro against the signatures contained in said databases. A user is alerted when a macro is detected with a signature that matches a signature from the first database and /or in the event the macro has a signature that matches a signature from the second and third databases. Thus, *Hyppönen* fails to disclose, teach, or suggest "monitoring the program instructions as they enter the DELI to determine if the program instructions have been previously cached within the DELI, wherein the determination of whether the program instructions have been cached is responsive to an association between native application code and analogues that have been cached within the code caches within the DELI." Consequently, the combination of *Sheymov* and *Hyppönen* does not render Applicants' dependent claim 5 obvious. Accordingly, the rejection of claim 5 should be withdrawn.

CONCLUSION

In light of the foregoing amendments and for at least the reasons set forth above, Applicants respectfully submit that all objections and/or rejections have been traversed, rendered moot, and/or accommodated, and that the now pending claims 1 – 24 are in condition for allowance. Favorable reconsideration and allowance of the present application and all pending claims are hereby courteously requested. If, in the opinion of the Examiner, a telephonic conference would expedite the examination of this matter, the Examiner is invited to call the undersigned attorney at (770) 933-9500.

Respectfully submitted,

THOMAS, KAYDEN, HORSTEMEYER & RISLEY, L.L.P.

Robert A. Blaha
Reg. No. 43,502

THOMAS, KAYDEN,
HORSTEMEYER & RISLEY, L.L.P.
100 Galleria Parkway N.W., Suite 1750
Atlanta, Georgia 30339
(770) 933-9500